

**IN THE CLAIMS:**

1. (Original) A method for managing resources of a physical processor, comprising:  
determining whether a first logical processor on the first physical processor is idle;  
determining whether a second logical processor on the first physical processor is busy if the first logical processor is idle; and  
relinquishing resources of the first physical processor to the second logical processor if the second logical processor is busy.
2. (Original) The method of claim 1, wherein the step of determining whether the first logical processor is idle comprises:  
determining whether the first logical processor is running a current job; and  
determining whether a first run queue corresponding to the first logical processor is empty if the first logical processor is not running a current job, wherein the first logical processor is idle if the first run queue is empty.
3. (Original) The method of claim 2, further comprising:  
running a job from the first run queue on the first logical processor if the first run queue is not empty.
4. (Original) The method of claim 2, wherein the first logical processor is not idle if the first logical processor is running a current job.
5. (Currently amended) The method of claim 1, further comprising:  
determining whether a job is available in a second run queue corresponding to a third logical processor on a second physical processor if the second logical processor on the first physical processor is not busy.

6. (Original) The method of claim 5, further comprising:  
running a job from the second run queue on the first logical processor if a job is available in the second run queue.
7. (Original) The method of claim 1, wherein the second logical processor consumes resources of the first physical processor if the first logical processor has a lowered priority.
8. (Currently amended) The method of claim 1, wherein the step of relinquishing the first physical processor resources comprises:  
lowering the priority of the first logical processor.
9. (Original) The method of claim 8, wherein the step of lowering the priority of the first logical processor comprises lowering the priority of the first logical processor for a predetermined time period.
10. (Original) The method of claim 9, further comprising raising the priority of the first logical processor after the predetermined period of time.
11. (Original) The method of claim 10, further comprising dispatching a job to the first logical processor in response to the raised priority.
12. (Currently amended) An apparatus for controlling ~~the~~ an active number of run queues on a first physical processor, comprising:  
first determination means for determining whether a first logical processor on the first physical processor is idle;  
~~first~~ second determination means for determining whether a second logical processor on the first physical processor is busy if the first logical processor is idle; and  
relinquishing means for relinquishing resources of the first physical processor to the second logical processor if the second logical processor is busy.

13. (Original) The apparatus of claim 12, wherein the first determination means comprises:  
means for determining whether the first logical processor is running a current job;  
and  
means for determining whether a first run queue corresponding to the first logical processor is empty if the first logical processor is not running a current job, wherein the first logical processor is idle if the first run queue is empty.
14. (Original) The apparatus of claim 13, further comprising:  
means for running a job from the first run queue on the first logical processor if the first run queue is not empty.
15. (Original) The apparatus of claim 13, wherein the first logical processor is not idle if the first logical processor is running a current job.
16. (Currently amended) The apparatus of claim 12, further comprising:  
means for determining whether a job is available in a second run queue corresponding to a third logical processor on a second physical processor if the second logical processor on the first physical processor is not busy.
17. (Original) The apparatus of claim 16, further comprising:  
means for running a job from the second run queue on the first logical processor if a job is available in the second run queue.
18. (Original) The apparatus of claim 12, wherein the second logical processor consumes the resources of the first physical processor if the first logical processor has a lowered priority.
19. (Original) The apparatus of claim 12 wherein the relinquishing means comprises:  
priority means for lowering the priority of the first logical processor.

20. (Original) The apparatus of claim 19, wherein the priority means comprises means for lowering the priority of the first logical processor for a predetermined time period.
21. (Original) The apparatus of claim 20, further comprising means for raising the priority of the first logical processor after the predetermined period of time.
22. (Original) The apparatus of claim 21, further comprising means for dispatching a job to the first logical processor in response to the raised priority.
23. (Currently amended) A computer program product, in a computer readable medium, for controlling the an active number of run queues on a first physical processor, comprising:
- instructions for determining whether a first logical processor on the first physical processor is idle;
  - instructions for determining whether a second logical processor on the first physical processor is busy if the first logical processor is idle; and
  - instructions for lowering the priority of the first logical processor if the second logical processor is busy.